EE314 Digital Electronics Laboratory 2017-2018 Spring Term Project Final Report An FPGA Based Oscilloscope

1st Halil TEMURTAS *2094522* *halil.temurtas@metu.edu.tr*
2nd Erdem TUNA *2167419* *erdem.tuna@metu.edu.tr*

***Abstract*—The design of an FPGA based digital oscilloscope**

***Index Terms*—FGPA, oscilloscope, ADC, VGA**

I. Introduction

In this project, it is intended to realize a digital oscilloscope by using an FPGA. The FGPA will receive an analog signal, digitize it through an ADC; then will calculate the required parameters. Lastly data and input signal will be displayed on a VGA screen. The overall diagram is shown in *Figure 1*. The implementation logics and algorithms are discussed in respective sections.

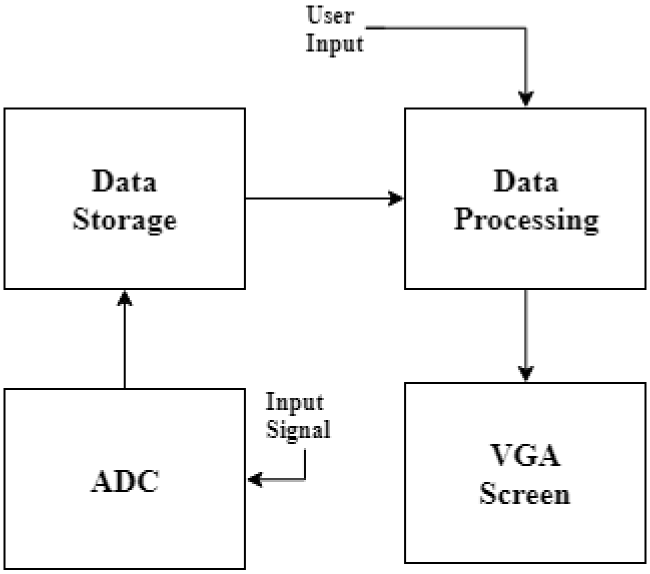


Figure 1: The Block Diagram of the Project

II. ALTERA DE1-SoC

In this project, we have used ALTERA DE1_SoC Development Kit for main unit and a VGA Monitor as a screen for the oscilloscope. In this part, our aim is to introduce the FPGA used in the project. The overall look of the device can be seen at *Figure 2*, as can be seen from the figure, the Development Kit consists of multiple parts aside from FPGA. In this project, GPIO Pins, seven segment display, push buttons, switches and VGA output of DE1-SoC was used.

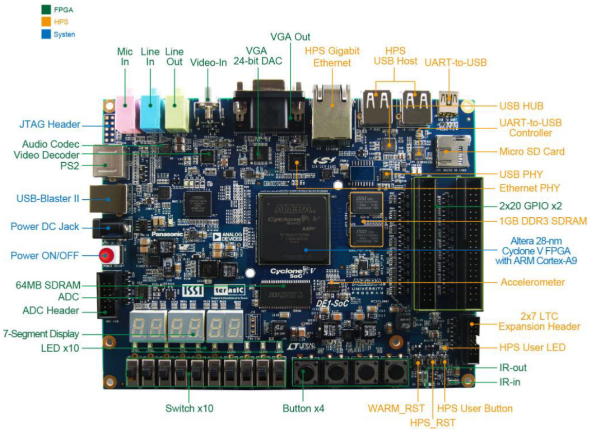


Figure 2: ALTERA DE1_SoC Development Kit

GPIO pins, also known as general-purpose input/output pins, were used for getting the analog input that is desired to be monitored through oscilloscope. As can be seen from the *Figure 3*, the basic circuitry for this pins includes Schottky diodes for protection, and a resistor.

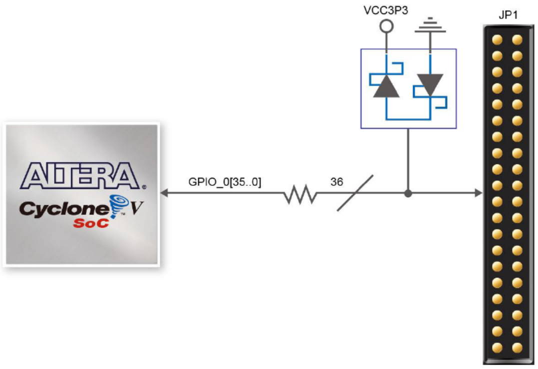


Figure 3: GPIO Pins

Seven segment display was used for testing the output values of computation value without using external monitor that we had some difficulties to find. As can be seen from the *Figure 4*, every led on the seven segment is connected through a resistor to the FPGA.

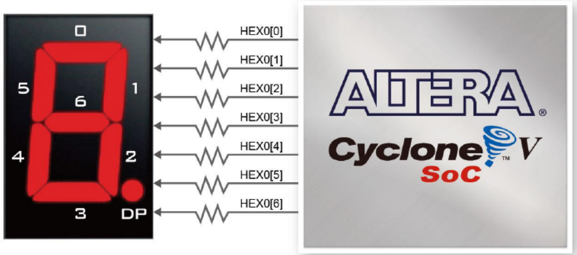


Figure 4: Seven Segment Display

Finally, the switch buttons were used as a mode buttons of the oscilloscope and push buttons were used as a division changer for the oscilloscope. The connections to FPGA can be seen at *Figure 5* and *Figure 6*. Finally the VGA connection used will be explained later in the report.

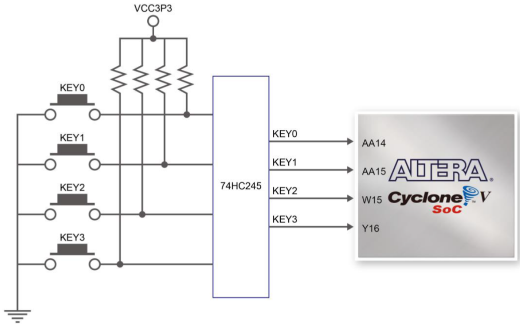


Figure 5: Push Buttons

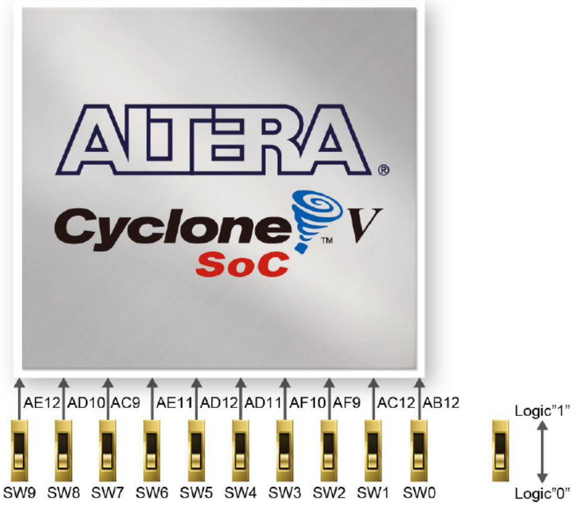


Figure 6: Switch Buttons

III. Modules

The modules and functional blocks designed and used in the project are presented in this section.

A. ADC

This module quantizes the analog input data. The hardware used is LTC2308 that is built into the FPGA. The Altera provides an example code regarding the use of the built-in ADC[1]. By evaluating the provided functionalities of this example with the help of a MsC project [2], a code is written in Verilog HDL and in Qsys environment. Qsys instantiates the internal connections of embedded modules to use ADC controller. The resulting Qsys module is exported as BSF file and that is shown in *Figure 7*.

This module couldn't be used in the project since the VGA screen wasn't driven. It wasn't possible to test the output of the ADC module. Also, integration of the module with the rest of the project couldn't be done.

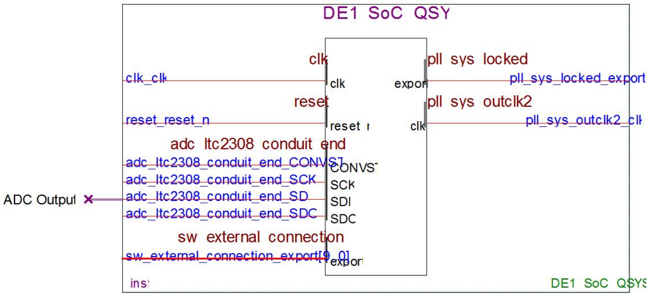


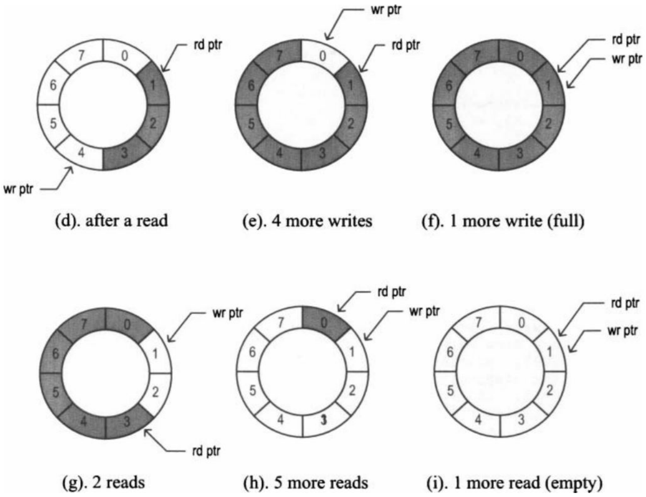
Figure 7: The Block Diagram of the ADC

B. RAM

The main function of RAM is to introduce a block to the system that can be both writable and readable. Writing is necessary to save data coming from ADC and reading is also necessary to process the data and show them on VGA screen. The RAM should work with FIFO structure to realize proper operation on VGA screen. ”A FIFO (first-in-first-out) buffer is an ”elastic” storage between two subsystems.”[3]. A FIFO based read and write operation is depicted in *Figure 8*. The FIFO is designed and written, the resulting circuit block is shown in *Figure 8*. The implemented FIFO has 8 bits of stack height and 12 bits of stack width. It also indicates when stacks are half-filled or full-filled.

Throughout the project, the RAM unit functions as a memory of the system. Firstly, the data coming from the output of the ADC is written to the RAM and secondly, the necessary calculations are made by calculation unit and written the data again. Finally, with a proper frequency, the data is read by VGA controller in order to reflect the wanted signal to the monitor. If this part is not working properly, the overall project can not operate properly.

In the demo, this module wasn't shown due to incompleteness of the rest of the project, even though FIFO was working properly.



(d). after a read (e). 4 more writes (f). 1 more write (full)

(g). 2 reads (h). 5 more reads (i). 1 more read (empty)

Figure 8: FIFO Working Principle[3]

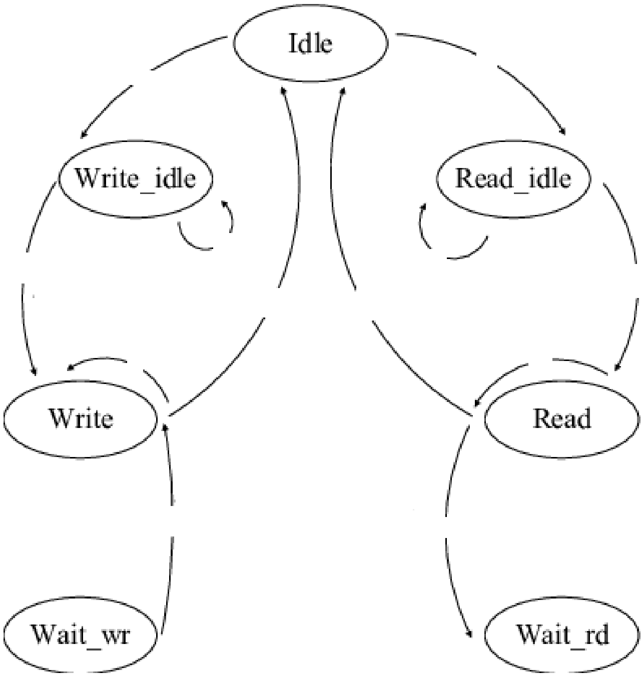


Figure 9: State Diagram for RAM

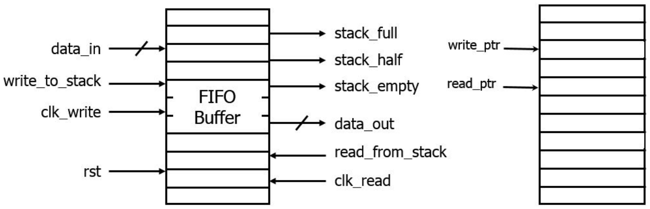


Figure 10: FIFO Block

C. Computation Unit

Computation unit is the unit responsible for all kinds of mathematical calculation of the project. For instance mean value calculation for the input signal are conducted in this unit. This Unit can be considered as a brain of the project. Some important parts are as follows,

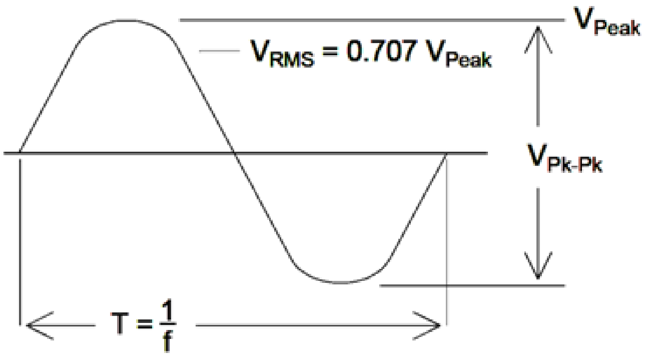


Figure 11: Voltage Waveform and its Properties

1) Calculation of Peak-to-Peak Value :

To find the peak-to-peak value of the input voltage, we needed to find the maximum and minimum voltage values of the given input. For that purpose we have used a well known sorting algorithm. Initially a point on the waveform was picked and every point next to it was compared to it. If the next point on the voltage is bigger than the initially stored value, the stored value is replaced with that value. After tracing every point on the waveform, the biggest value can be found. Similarly, the lowest value on the voltage waveform can be found.

After finding minimum and maximum points at the voltage waveform, peak-to-peak value is the difference between these values as can be seen from the *Figure 11*. Basic state diagram for this module can be seen at *Figure 12*. ”0” refers the situation that the next bit compared is not bigger than the already stored value. ”1” refers the opposite situation.

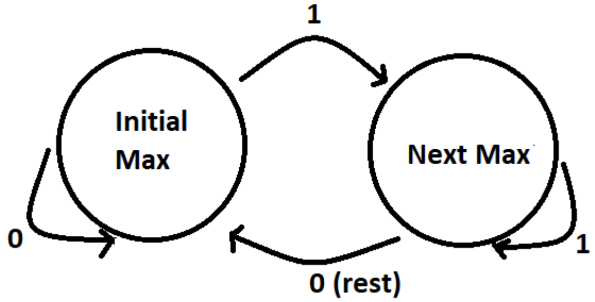


Figure 12: State Transition Diagram for Peak-to-Peak Calculator

2) Calculation of Mean Value:

Mean value calculation is another essential part of the project since the requirements states that the voltage waveform needed to start from the mean value of the waveform. It should also be noted that mean value of the voltage waveform can sometimes be confused with the rms value of the same voltage waveform. However, since we deal with only the symmetrical waveforms as an input, the calculation is rather easy and do not include complex calculations like integration over some period. Assuming a symmetrical sinusoidal voltage waveform as in *Figure 13* the mean or average value of the voltage waveform can be calculated easily as follows:

$$v(t) \;=\; A + Bsin(2\pi ft)$$

$$A + B \;=\; Maximum\ value\ of\ the\ voltage$$

$$A - B \;=\; Minimum\ value\ of\ the\ voltage$$

$$V_{mean} \;=\; \frac{(A+B)+(A-B)}{2} \;=\; A\ \ Volts$$

And, the peak-to-peak voltage can be found to be as;

$$V_{P-P} \;=\; 2B\ \ Volts$$

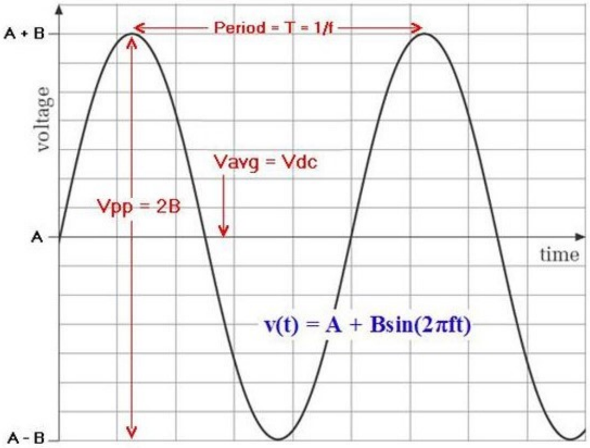


Figure 13: A Sinusoidal Voltage Wave[5]

3) Calculation of DC Offset:

DC offset value can be summarized as a mean amplitude displacement from the ground level. Assuming that we knew the ground level and calculated the mean value of the voltage, we could simply find the DC Offset Value that is equal to the mean value of the voltage.

For example, for the voltage waveform at *Figure 13*, the $A\ Volts$ is also the DC Offset of the voltage waveform.

The same methodology can be applied for the square and triangular waves also.

4) Calculation of Frequency:

Measuring the frequency information of the voltage waveform is one of the most important functions of the digital oscilloscopes in the market. Since the project was all about building oscilloscope, it would be meaningless not to put frequency information on the screen. For that purpose, we have used the mean value information that we have calculated in the computation unit. Tracing the waveform from the first intersection of the Mean Value level and the waveform itself, the second intersection gives an information about half period and the third information gives an information about the full period.

Considering another sinusoidal waveform at *Figure 14* and assuming that the first intersection occurs at the origin. The frequency information can be found as follows.

$$Y - 0 \;=\; Y\ is\ equal\ to\ the\ period\ T$$

$$frequency\ = f\ = \frac{1}{T}\ = \frac{1}{Y}\ \ Hz$$

The same methodology can be applied for the square and triangular waves also.

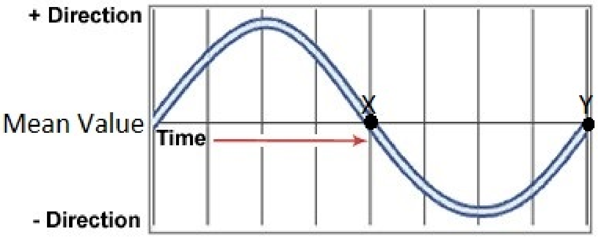


Figure 14: Another Sinusoidal Voltage Waveform[6]

5) Mode Selection (AC/DC):

AC/DC Modes are one of the most fundamental mods of digital oscilloscopes in market. In this part we will build a module to build our own selection mode. One slide switch will be assigned to retrieve the desired mode information from the user. Based on this information, screen will display the input signal either with the DC offset or without the DC offset. Basic state diagram for this module can be seen at *Figure 15*.

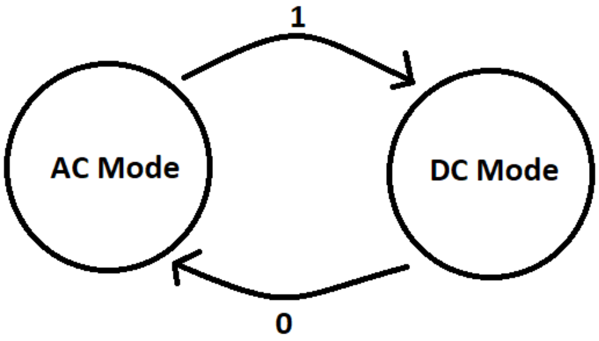


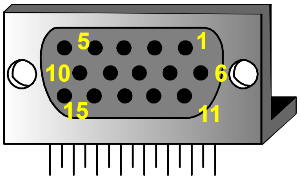
Figure 15: State Diagram for the Mode Selection

AC Mode: In AC Mode operation of the oscilloscope, the DC offset voltage is removed from the input voltage before it is reflected to the VGA monitor. For that the DC offset data found earlier will be removed from stored data.

DC Mode: In DC Mode operation of the oscilloscope, the DC offset voltage is untouched from the stored data of the input voltage. The stored data is reflected directly to the VGA monitor.

D. VGA Screen

VGA is a widely used standard in video industry for the transmission of video signals from a computer or microprocessor into a monitor or TV. The input pin configuration for the VGA Monitor can be seen at *Figure 16*. VGA provides 640x480 pixel resolution. This resolution, however, is not the total pixel in horizontal and vertical axes. There is a blank border frame surrounding the display area. The horizontal axis has 48 pixels of border width on the left and 16 pixels of border width on the right side of the screen. Additionally, there are 96 pixels of border width for ray tube the retrace through the diagonal of the screen. The vertical axis has also similar additional pixels. 10 and 33 pixels for top and bottom borders, respectively. Also 2 pixels for the vertical retrace of the ray tube. The total structure of a VGA screen can be seen in *Figure 18*. The whole screen is 800x525 pixels.



Pin 1: Red	Pin 5: GND
Pin 2: Grn	Pin 6: Red GND
Pin 3: Blue	Pin 7: Grn GND
Pin 13: HS	Pin 8: Blu GND
Pin 14: VS	Pin 10: Sync GND

Figure 16: VGA Input Pins

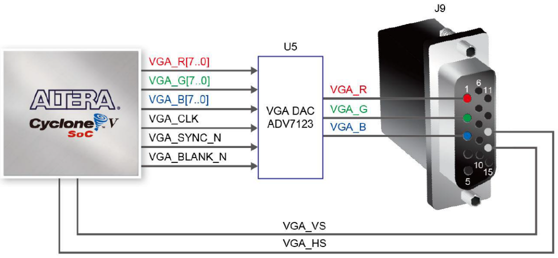


Figure 17: The Block Diagram of the Project

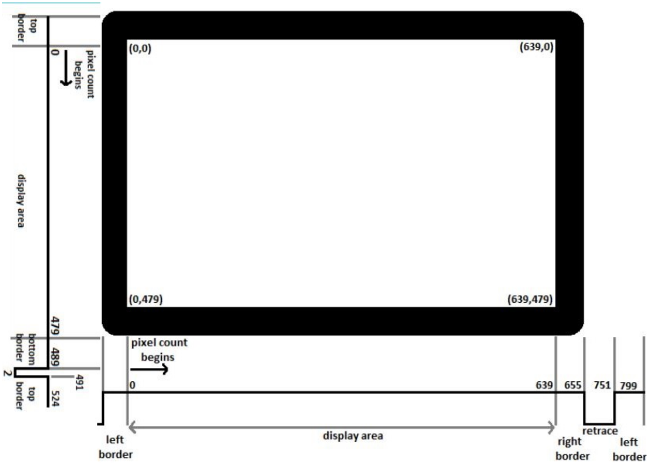


Figure 18: A Frame of VGA Screen with Control Signals

1) VGA Controller:

The VGA controller administrates the digitally stored input signal and the measured properties of the signal to display them on the screen. The two counters in this module, named ”h_count” and ”v_count” are to keep track of the vertical and horizontal pixels. Also there are two more counters to synchronize the display area, blank are and the retrace area of the screen. These counters are ”h_sync” and ”v_sync” and they are the output signals of PINs 13 and 14 of VGA output with an active low behaviour. Lastly, ”o_red”, ”o_blue” and ”o_green” are the 8-bit output bus signals determine the color of the respective pixel by mapping them on 255,255,255 color scale. The resulting diagram showing the signals of the VGA Controller is shown in *Figure 19* [4].

The waveform, frequency, peak-to-peak voltages of the signals are dynamic measurements, while texts indicating them are not. For this reason, with the reference of [2], static images can be shown in ”.MIF” formats for simplicity. These images are first saved as ”.BMP” then converted to ”.MIF” and stored in a simple ROM structure.

The codes are written alongside with the aforementioned ideas, however, the display on the VGA screen could not be achieved.

combines the data from RAM and Computation Unit to create a signal that can be displayed on the VGA monitor. Each of the RAM Modules contains an image that is ready for display on the screen. However, the data must be positioned relative to each other and combined. Also this module performs once a second as desired in the project requirements. The VGA controller also gets data from different data inputs such as Time/div and Voltage/div in order to reflect the waveform as user requires. In this part multiple clock signals and counters needed to display the waveform accordingly. Basic VGA Controller can be seen at *Figure 19*.

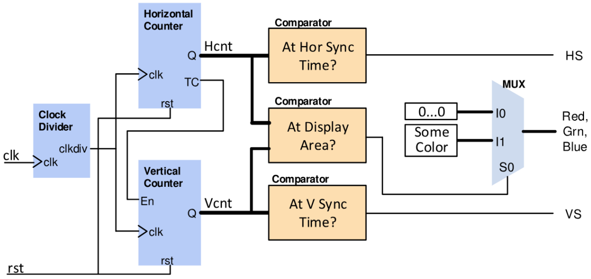


Figure 19: Internal Structure of the VGA Controller

IV. Conclusion

In this project, the aim was to design a FPGA based digital oscilloscope using Verilog. The project consisted of four main part that are Analog to Digital Converter, RAM, Computation Unit and VGA Screen. Overall diagram of the project can be seen at *Figure 1*. None of the parts were implemented properly. This is due to lack of knowledge in Verilog, RAM structure, internal flow of data in FPGA and computer structure. It would be better to implement some parts of the project as the project but the project itself was too complex to build for us. The importance of Verilog is understood very well.

References

[1] “De1-Soc CD.” [Online]. Available: http://download.terasic.com/downloads/cd-rom/de1-soc/DE1-SoC_v.5.0.1_HWrevF_SystemCD.zip. [Accessed: 21-May-2018].

[2] “Digital Scope Implemented on Altera DE1-SoC.” [Online]. Available: https://people.ece.cornell.edu/land/courses/eceprojectsland/STUDENTPROJ/2015to2016/hj424/hj424_report_201605191237.pdf. [Accessed: 21-May-2018].

[3] P.P. Chu, Embedded SoPC Design with NIOS II Processor and Verilog Examples. New Jersey, 2012.

[4] ”VGA Display Controller” [Online]. Avaible : https://learn.digilentinc.com/Documents/269

[5] [Online]. Avaible : http://images.slideplayer.com/19/5725931/slides/slide_2.jpg

[6] [Online]. Avaible : http://slideplayer.com/slide/5725931/